

JUN 09 2006

TRANSMITTAL OF APPEAL BRIEF (Large Entity)

Docket No.  
YOR920000196US1

In Re Application Of: **Dimitri Kanevsky, et al.**

Application No.	Filing Date	Examiner	Customer No.	Group Art Unit	Confirmation No.
09/584,810	May 31, 2000	Alina A. Boutah	23389	2143	2276

Invention: **UNIVERSAL CONVERSION SERVER**

COMMISSIONER FOR PATENTS:

Transmitted herewith is the Appeal Brief in this application, with respect to the Notice of Appeal filed on:

The fee for filing this Appeal Brief is: **\$500.00**

- ☐ A check in the amount of the fee is enclosed.
- ☒ The Director has already been authorized to charge fees in this application to a Deposit Account.
- ☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 50-0510/IBM. I have enclosed a duplicate copy of this sheet.
- ☐ Payment by credit card. Form PTO-2038 is attached.

**WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.**

*John S. Sensny*  
Signature

Dated: **June 5, 2006**

**John S. Sensny**  
Registration No. 28,757

**Scully, Scott, Murphy & Presser, P.C.**  
400 Garden City Plaza - Suite 300  
Garden City, New York 11530  
(516) 742-4343

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on

June 5, 2006

(Date)

*John S. Sensny*  
Signature of Person Mailing Correspondence

**John S. Sensny**

Typed or Printed Name of Person Mailing Correspondence

cc: JSS:jy



**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

**Applicants:** Dimitri Kanevsky, et al.

**Examiner:** Alina A. Boutah

**Serial No:** 09/584,810

**Art Unit:** 2143

**Filed:** May 31, 2000

**Docket:** YOR920000196US1 (13539)

**For:** UNIVERSAL CONVERSION SERVER

**Dated:** June 5, 2005

**Confirmation No.:** 2276

Hon. Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF**

Sir:

Pursuant to 35 U.S.C. 134 and 37 C.F.R. 41.37, entry of this Appeal Brief in support of the Notice of Appeal filed April 3, 2006, in the above-identified matter is respectfully requested.

---

**CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)**

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Hon. Commissioner for Patents, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450, on June 5, 2006.

Dated: June 5, 2006

John S. Sensny  
John S. Sensny

06/09/2006 RFEKADU1 00000021 500510 09584810  
01 FC:1402 500.00 DA

**I. Statement of Real Party in Interest**

The real party in interest in the above-identified patent application is the International Business Machines Corporation.

**II. Statement of Related Appeals and Interferences**

There are no other prior or pending appeals, interferences or judicial proceedings known to appellants, the appellants' legal representative, or assignee which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

**III. Status of Claims**

**A. Claim Status**

Claim 1 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 2 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 3 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 4 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 5 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 7 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 8 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 9 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 10 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 11 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 12 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 13 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 14 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 15 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 16 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 17 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 18 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

Claim 19 stands rejected based on 35 U.S.C. §103 as being unpatentable over U.S. Patent 6,092,114 (Shaffer, et al.) in view of U.S. Patent 6,549,918 (Probert, Jr, et al.) and further in view of a document titled "Conversion Service" (CERN).

## **B. Appealed Claims**

Claims 1-5 and 7-19 are appealed. A clean copy of these claims is contained in Appendix A to this Appeal Brief.

#### **IV. Status of Amendments**

An Amendment dated February 2, 2006 was filed subsequent to the final rejection. That Amendment has been entered.

#### **V. Summary of Claimed Subject Matter**

##### **Concise explanation of the subject matter of claim 1**

Claim 1 defines a method for re-formatting computer files. This method comprises the steps of inputting a data file into a computer 101 (page 4, line 21) having a specified operating system, and using that computer 101 to determine if the data file is compatible with the specified operating system. If the data file is not compatible with the computer 101, then the computer transmits the data file over the Internet 100 from the computer to a universal server 104. The universal server then transforms the data file into a format compatible with the specified operating system of the computer (Page 5, line 20-Page 6, line 22; Page 7, line 30-Page 8, line 21), and sends the transformed data file back to the computer 101.

##### **Concise explanation of the subject matter of Claim 7**

Claim 7 is directed to a universal program conversion method. In accordance with this method, data is entered into a computer 101 (Page 4, line 21), which has a specified operating system, and the computer 101 checks to determine whether the format of the entered data is compatible with the specified operating system(OS) in the computer. If the format is not compatible with this operating system, the computer 101 sends the data from the computer over a network 100 to a remote Universal Driver 109 (Page 6, line 24-Page 7, line 20), which reformats the data into a format that is compatible with the specified OS. The Universal Driver 109 checks to determine if the source code exists on a storage 110 of source code (Page 7, line 30-Page 8, line 7). If the source code exists on this storage 110, the Universal Driver

then recompiles the data in a new OS, and the Universal Driver 109 then sends the data to the user.

Concise explanation of the subject matter of Claim 8

Claim 8 defines a system for re-formatting computer files. Generally, this system comprises a computer 101 (Pge 4, line 21) and a universal server 104. The computer has input means 102, 103 (Page 4, lines 22 and 23) for receiving a data file, a specified operating system, and means 101 (Page 10, lines 10-13) for determining if the data file is compatible with the specified operating system. The computer also includes means 101 (Page 10, lines 13-15) for transmitting the data file over the Internet from said computer to the universal server 104, if the data file is not compatible with the specified operating system of the computer. The universal server 104 includes means (Page 5, line 20-Pge 6, line 22; Page 7, line 30-Pge 8, line 21) for transforming the data file into a format compatible with the specified operating system of the computer 101, and means 104 for sending the transformed data file back to the computer.

Concise explanation of the subject matter of Claim 12

Claim 12 is directed to a program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for re-formatting computer files. These method steps comprise inputting a data file into a computer 101 (Page 4, line 21) having a specified operating system, and using that computer 101 (Page 10, lines 10-13) to determine if the data file is compatible with the specified operating system of the computer.

If the data file is not compatible with the computer 101, the computer is used to transmit the data file over the Internet 100 from the computer to a universal server 104 (Page 5, lines 1 and 2). The universal server 104 then transforms the data file into a format compatible with the specified operating system of the computer 101 (Page 5, line 20-Pge 6, line 22; Page 7, line 30-Pge 8, line 21), and sends the transformed data file back to the computer 101.

Identification of means plus function and step plus function under 35 U.S.C. 112, and identification of the structure, materials or acts described in the specification as corresponding to each claimed function.

Claim 1

Claim 1 includes the following step plus functions

1. inputting a data file into a computer  
corresponding structure, material or acts

This may be done via a driver 110, voice commands, a typed command or the Internet (Page 4, lines 23-30), or by entering files (Page 10, lines 6 and 7).

2. using the computer to determine if the data file is compatible with the operating system

corresponding structure, material or acts

A check is made to determine whether the file format is compatible with the operating system in the computer (Page 10, lines 8-10).

3. transmitting the data file over the Internet from the computer to a universal server

corresponding structure, material or acts

the data file is transmitted over the Internet to the universal server (Page 2, line 31 and 32; Page 4, line 32-Page 5, line 2).

4. transforming the data file into a format compatible with the specific operating system

corresponding structure, material or acts

This can be done by re-formatting the data file (Page 6, lines 2-22), by finding a counterpart file in a suitable format (Page 7, lines 16-28), or by re-compiling the source code of the file (Page 7, line 30-Page 8, line 21).

5. sending the transformed data back to the computer

corresponding structure, material or acts



This is done by the Universal Server (Page 11, lines 10, 11, 25 and 26).

#### Claim 7

Claim 7 includes the following step plus functions

1. entering the data into a computer

corresponding structure, material or acts

This may be done via a driver 110, voice commands, a typed command or the Internet (Page 4, lines 23-30), or by entering files (Page 10, lines 6 and 7).

2. checking to determine if the data file is compatible with the operating system

corresponding structure, material or acts

A check is made to determine whether the file format is compatible with the operating system in the computer (Page 10, lines 8-10).

3. sending the data from the computer over a network to a remote universal driver

corresponding structure, material or acts

the data file is transmitted over the Internet to the universal driver (Page 10, lines 13 and 14).

4. reformatting the data file into a format compatible to the specified OS

corresponding structure, material or acts

This can be done by re-formatting the data file (Page 6, lines 2-22), by finding a counterpart file in a suitable format (Page 7, lines 16-28), or by re-compiling the source code of the file (Page 7, line 30-Page 8, line 21).

5. sending the data to a universal formatting server

corresponding structure, material or acts

This is done by the Universal Driver (Page 6, lines 13-15).

6. checking to determine whether it is necessary to reformat the data

corresponding structure, material or acts

This is done by the Universal Driver (Page 10, lines 8-15).

7. processing the data as the user requests

corresponding structure, material or acts

This is done by the Universal Driver (Page 10, lines 26-29).

8. sending the data to the universal server

corresponding structure, material or acts

This is done by the Universal Driver (Page 10, lines 29 and 30).

9. checking whether the data are executable

corresponding structure, material or acts

This is done by the Universal Server (Page 10, lines 30-32).

10. checking the Universal Driver to determine whether the data can be formatted on the Universal Driver.

corresponding structure, material or acts

This is done by the Universal Driver (Page 11, lines 2-7).

11. formatting the data at the Universal Driver.

corresponding structure, material or acts

This is done by the Universal Driver (Page 11, lines 21-24).

12. sending the formatted data to the user

corresponding structure, material or acts

The Universal Driver does this (Page 11, lines 25 and 26).

13. checking to determine if the source code exists on a storage of source code.

corresponding structure, material or acts

The Universal driver does this (Page 11, lines 9-11).

14. recompiling the data

corresponding structure, material or acts

The Universal Driver does this (Page 11, lines 9-11).

15. sending the data to the user.

corresponding structure, material or acts

The Universal Driver does this (Page 11, line 11).

16. checking for instructions to format data.

corresponding structure, material or acts

The Universal Driver does this (Page 11, lines 21 and 22).

17. formatting the data according to the instructions

corresponding structure, material or acts

The Universal Driver does this (Page 11, lines 24 and 25).

18. sending the data to the user.

corresponding structure, material or acts

The Universal Driver does this (Page 11, lines 25 and 26).

#### Claim 8

Claim 8 includes the following means plus functions

1. input means for receiving a data file.

corresponding structure, material or acts

This may be done via a driver 110, voice commands, a typed command or the Internet (Page 4, lines 23-30), or by entering files (Page 10, lines 6 and 7).

2. means for determining if the data file is compatible with the specified operating system of the computer.

corresponding structure, material or acts

Computer 101 does this by checking to determine whether the file format is compatible with the operating system in the computer (Page 10, lines 8-10).

3. means for transmitting the data file over the Internet from said compute to the universal server.

corresponding structure, material or acts

Computer 101 does this by transmitting the data file over the Internet to the universal server (Page 2, line 31 and 32; Page 4, line 32-Page 5, line 2).

4. means for transforming the data file into a format compatible with the specified operating system of the computer.

corresponding structure, material or acts

The Universal Server does this by re-formatting the data file (Page 6, lines 2-22), by finding a counterpart file in a suitable format (Page 7, lines 16-28), or by re-compiling the source code of the file (Page 7, line 30-Page 8, line 21).

5. means for sending the transformed data file back to the computer.

corresponding structure, material or acts

This is done by the Universal Server (Page 11, lines 10, 11, 25 and 26).

#### Claim 12

Claim 12 includes the following step plus functions

1. inputting a data file into a computer

corresponding structure, material or acts

This may be done via a driver 110, voice commands, a typed command or the Internet (Page 4, lines 23-30), or by entering files (Page 10, lines 6 and 7).

2. using the computer to determine if the data file is compatible with the operating system

corresponding structure, material or acts

A check is made to determine whether the file format is compatible with the operating system in the computer (Page 10, lines 8-10).

3. transmitting the data file over the Internet from the computer to a universal server  
corresponding structure, material or acts  
the data file is transmitted over the Internet to the universal server (Page 2, line 31 and 32; Page 4, line 32-Page 5, line 2).

4. transforming the data file into a format compatible with the specific operating system  
corresponding structure, material or acts  
This can be done by re-formatting the data file (Page 6, lines 2-22), by finding a counterpart file in a suitable format (Page 7, lines 16-28), or by re-compiling the source code of the file (Page 7, line 30-Page 8, line 21).

5. sending the transformed data back to the computer  
corresponding structure, material or acts  
This is done by the Universal Server (Page 11, lines 10, 11, 25 and 26).

## **VI. Grounds of Rejection to be Reviewed On Appeal**

Appellants ask that the following grounds be reviewed:

2. Whether Claims 1-5 and 7-19 are unpatentable under 35 USC §103 over Shaffer, et al, Probert, Jr. et al. and CERN.

## **VII. Argument**

### **A. Rejection of Claims 1-5, and 8-19.**

The rejection of Claims 1-5, 8-19 is respectfully traversed because the prior art does not disclose or suggest using a computer, first, to determine if a file is compatible with the computer's operating system, and second, using that same computer to transmit the data file

over the Internet to a Universal Server for conversion, as described in independent Claims 1, 8 and 12. Claim 1 is representative of Claims 8 and 12.

More specifically, Probert, Jr., et al. describes an operating system layer between software components or application programs that expect information to be in one format and a persistent store manager of the operating system that maintains the information in this persistent state. This operating system is used to provide "on the fly" transformation between the file format expected by the application layer and the format used by the persistent store manager. The express teaching of Probert, Jr. et al, thus, is to provide this conversion software on the computer, and not on a remote server.

Shaffer, et al. describes a method and system for exchanging electronic messages. With the process disclosed in Shaffer, et al, however, the data is not sent to a server by a computer that earlier received the file and determined that the file was not compatible with the computer's operating system. For example, the Abstract of Shaffer, et al. states that "If it is determined that a file format conversion is required, the conversion operation is assigned to a server..." Shaffer, et al. does not disclose that the computer, which has found the data file incompatible, sends the actual data file to the conversion server.

CERN was cited in the Office Action for its teaching of allowing users to upload files over the Internet from a user's station to be converted. Among other differences between the present invention and CERN, this reference does not disclose using the computer to determine that the file is not compatible with the computer's operating system. Instead, with the CERN procedure, the computer user determines what files to send for conversion.

Independent Claims 1, 8 and 12 clearly set forth the above-discussed difference between the present invention and the prior art. Specifically, Claims 1 and 12 describe the features that the computer determines whether the format of the file is compatible with the computer operating system, and if the data file is not compatible with the computer, that computer transmits that data file over the Internet from the computer to a universal server, which then transforms the file into a format compatible with the computer.

In this connection, it may be helpful to keep in mind that the Court of Appeals for the Federal Circuit has taught that:

“an ‘obvious to try’ standard is not a legitimate test of patentability...The statutory standard of 103 is whether the invention, considered as a whole, would have been obvious to one skilled in the art, not whether it would have been obvious to one skilled in the art to try various combinations.” N.V. Akzo v. E.I. duPont deNemours & Co., 1 USPQ2d 1704, 1707 (Fed Cir. 1987).

In light of the above-discussed differences between Claims 1, 8 and 12 and the prior art, and because of the advantages associated with those differences, it cannot be said that any of these claims is obvious in view of that prior art. Hence, Claims 1, 8 and 12 patentably distinguish over the prior art and are allowable. Claims 2-5 and 17-19 are dependent from Claim 1 and are allowable therewith. Also, Claims 9-11 are dependent from Claim 8 and are allowable therewith; and Claims 13-15 are dependent from, and are allowable with, Claim 12. The Board is thus respectfully requested to reverse the rejection of Claims 1-5 and 8-19 under 35 U.S.C. 103.

**B. Rejection of Claim 7 under 35 U.S.C. 103**

The rejection of Claim 7 is also respectfully traversed because the prior art does not disclose or suggest using a computer, first, to determine if a file is compatible with the computer’s operating system, and second, using that same computer to transmit the data file over a network to a remote Universal Driver for conversion to a suitable format, as described in independent Claim 7.

Claim 7 is thus directed to the use of the Universal Driver to convert the data file to a compatible format. Analogous to Claim 1, Claim 7 describes the feature that a computer

determines if data, entered into the computer, is compatible with the operating system in the computer. As described in Claim 7, if the format of that data is not compatible with that computer, that computer sends that data over a network to a remote universal driver, which then reformats that data into a format compatible with the computer.

For reasons analogous to those advanced above in connection with Claim 1, the prior art, including Shaffer, et al, Probert, Jr. et al. and CERN, does not disclose or suggest the use of the same computer for both of the above-discussed functions. Accordingly, Claim 7 also patentably distinguishes over the prior art and is allowable, and the Board is, accordingly, respectfully asked to reverse the rejection of Claim 7 under 35 U.S.C. 103.

### **C. Conclusion**

In view of the above-discussed differences between Claims 1, 7, 8 and 12 and Shaffer, et al, Probert, Jr. et al. and CERN, and because of the advantages associated with those differences, these claims patentably distinguish over the prior art. Claims 2-5 and 17-19 are dependent from Claim 1. Also, Claims 9-11 are dependent from Claim 8; and Claims 13-16 are dependent from Claim 12. Thus, the rejection of the claims under 35 U.S.C. 103 is not proper, and the Board is requested to reverse this rejection.

### **VIII. Claims Appendix**

A clean copy of Claims 1-19 is contained in Appendix A to this Appeal Brief.

### **IX. Evidence Appendix**

Appellants are not relying on any affidavits, extrinsic documents or extrinsic evidence.

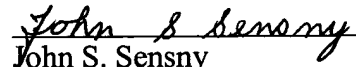


**IX. Related Proceedings Appendix**

As indicated above, there are no other prior or pending appeals, interferences or judicial proceedings known to appellants, the appellants' legal representative, or assignee which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

Respectfully submitted,

Dated: June 5, 2006

  
John S. Sensny  
Registration No. 28,757  
Attorney for Applicants

Scully, Scott, Murphy & Presser, P.C.  
400 Garden City Plaza – Suite 300  
Garden City, New York 11530  
(516) 742-4343

JSS:jy/gc  
Enclosure: Appendix A

## APPENDIX A

Claim 1. A method for re-formatting computer files, comprising the steps:

inputting a data file into a computer having a specified operating system;

using said computer to determine if the data file is compatible with the specified operating system;

if the data file is not compatible with the computer, said computer transmitting the data file over the Internet from said computer to a universal server; and

the universal server, transforming the data file into a format compatible with the specified operating system of the computer, and sending the transformed data file back to the computer.

Claim 2. A method according to Claim 1, wherein the transforming step includes the steps of, the universal server reading information from said computer; and on the basis of said information, identifying the type of file, and transforming the file into a different format of the same type.

Claim 3. A method according to Claim 1, further comprising the steps of:

a user of the computer identifying user requirements; and

transmitting the user requirements to the universal server; and wherein

the transforming step includes the step of re-formatting the file in accordance with the user requirements.

Claim 4. A method according to Claim 1, wherein, when data needs to be converted, the data are sent to a universal conversion server; the universal conversion server checks user requirements; if the universal conversion server finds that the service cannot convert a certain file, the service looks in a computer description; the computer description can be located on the computer or on a universal conversion server database.

Claim 5. A method according to Claim 1, wherein the data file is a computer program, and the transforming step includes the step of the universal server looking over the program to identify components of the program including links to the program source code, the program's executable code, the program's file name; entering data to a database of source codes, where many source codes are held; and if the same name exists among more than one program in the database, the Universal Server reads the information from a description module.

Claim 6 (Cancelled).

Claim 7. A universal program conversion method, comprising the steps:

entering data into a computer;

said computer having a specified operating system and checking to determine whether the format of the data is compatible with the specified operating system(OS) in the computer;

if the format is not compatible, said computer sending the data from the computer over a network to a remote Universal Driver;

on the Universal Driver, reformatting the data into a format compatible to the specified OS;

after the reformatting step, sending the data to a universal formatting server, to be converted to a format suitable for the user;

if it is determined that the data are compatible with the operating system, then checking to determine whether it is necessary to reformat the data;

if the data do not need to be reformatted, processing the data as the user requests; and otherwise, sending the data to the universal server; and this server checking whether the data are executables;

if the data are executables, then checking the Universal Driver to determine whether the data can be formatted on the Universal Driver; if the data can be so formatted, then formatting the data at the Universal Driver; and then sending the formatted data to the user; if the data can not be formatted at the Universal Driver, then checking to determine if the source code exists on a storage of source code; if the source code exists, the Universal Driver then recompiling the data in a new OS, and the Universal Driver then sending the data to the user; checking for instructions to format data; after the checking step, formatting the data are formatted according to the instructions, and then sending the data to the user.

Claim 8. A system for re-formatting computer files, comprising:

a computer having input means for receiving a data file;

said computer including a specified operating system and means for determining if the data file is compatible with the specified operating system of the computer; and

a universal server for reformatting data;

said computer including means for transmitting the data file over the Internet from said computer to the universal server, if the data file is not compatible with the specified operating system of the computer; and

wherein the universal server includes means for transforming the data file into a format compatible with the specified operating system of the computer, and means for sending the transformed data file back to the computer.

Claim 9. A system according to Claim 8, wherein the transforming means includes means for identifying the type of file, and for transforming the file into a different format of the same type.

Claim 10. A system according to Claim 8, wherein a user of the computer identifies user requirements; and the system further comprises:

means for transmitting the user requirements to the universal server; and wherein

the transforming means includes means for re-formatting the file in accordance with the user requirements.

Claim 11. A system according to Claim 8, wherein, when data needs to be converted, the data are sent to a universal conversion server; the universal conversion server checks user requirements; if the universal conversion server finds that the service cannot convert a certain file, the service looks in a computer description; the computer description can be located on the computer or on a universal conversion server database.

Claim 12. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for re-formatting computer files, the method steps comprising:

inputting a data file into a computer having a specified operating system;

using said computer to determine if the data file is compatible with the specified operating system of the computer;

if the data file is not compatible with the computer, ~~transmitting~~ using said computer to transmit the data file over the Internet from the computer to a universal server; and

the universal server, transforming the data file into a format compatible with the specified operating system of the computer, and sending the transformed data file back to the computer.

Claim 13. A program storage device according to Claim 12, wherein the transforming step includes the steps of, the universal server reading information from said computer; and on the basis of said information, identifying the type of file, and transforming the file into a different format of the same type.

Claim 14. A program storage device according to Claim 12, further comprising the steps of:

a user of the computer identifying user requirements; and

transmitting the user requirements to the universal server; and wherein

the transforming step includes the step of re-formatting the file in accordance with the user requirements.

Claim 15. A program storage device according to Claim 12, wherein, when data needs to be converted, the data are sent to a universal conversion server; the universal conversion server checks user requirements; if the universal conversion server finds that the service cannot convert a certain file, the service looks in a computer description; the computer description can be located on the computer or on a universal conversion server database.

Claim 16. A program storage device according to Claim 12, wherein the data file is a computer program, and the transforming step includes the step of the universal server looking over the program is looked over to identify components of the program including links to the program source code, the program's executable code, the program's file name; entering data to a database of source codes, where many source codes are held; and if the same name exists among more than one program, then the Universal Server reads the information from the description module.

Claim 17. A method according to Claim 1, further comprising the step of providing the Universal Server with access to a module having a series of source codes, and wherein the step of formatting the data file into a format compatible with the operating system of the computer modules includes the steps of:

the Universal Server obtaining from said module the source code for the data file; and

the Universal Server recompiling the data file, using the source code obtained from said module, into the format compatible with the operating system of the computer.

Claim 18. A method according to Claim 17, wherein the step of the Universal Server recompiling the data file includes the steps of, the Universal Server

using the source code obtained from said module to modify the source code of the data file; and

using a compiler to compile a new data file, compatible with computer, from the modified source code of the data file.

Claim 19. A method according to Claim 18, further comprising the step of the Universal Server reading from the computer the type of operating system on the computer.